

Figure 1 - Component Design Approach



Payment Component

Micro Structure Definition:

Internal Instance Variables

date:

Simple Date

currency:

Simple Symbol Simple Float

notional:

accrual:

Accrual Interface

Interface Definitions:

- 1) Accrual Interface supports:
 - start date
 - end date
 - rate
 - day count
 - processing interface
- 2) Processing interface (required)
 - valueEventInProcessor:

Figure 2 - Micro Structure of a Payment Component

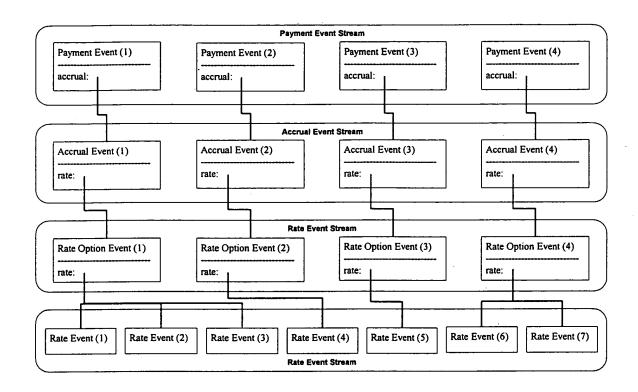


Figure 3 - Macro Structure Example

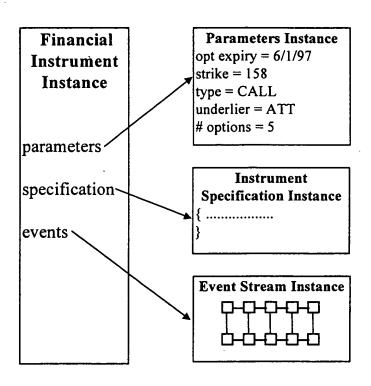


Figure 4 - Sample Financial Instrument Structure: Equity Option

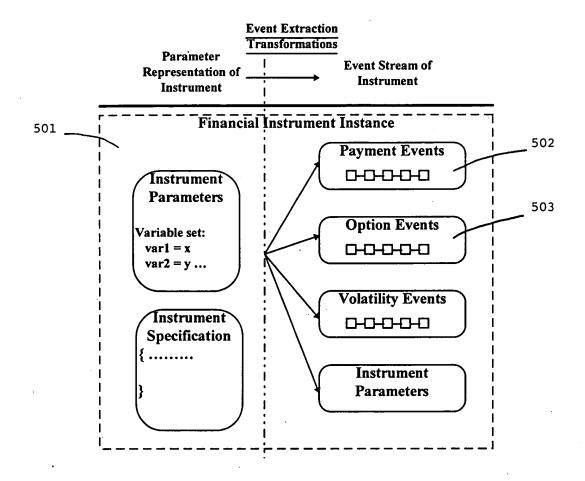


Figure 5 - Financial Instrument Structure Relationships

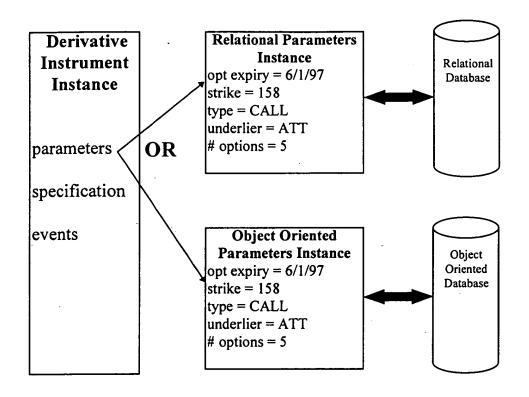


Figure 6 - Alternative Instrument Parameters Implementations

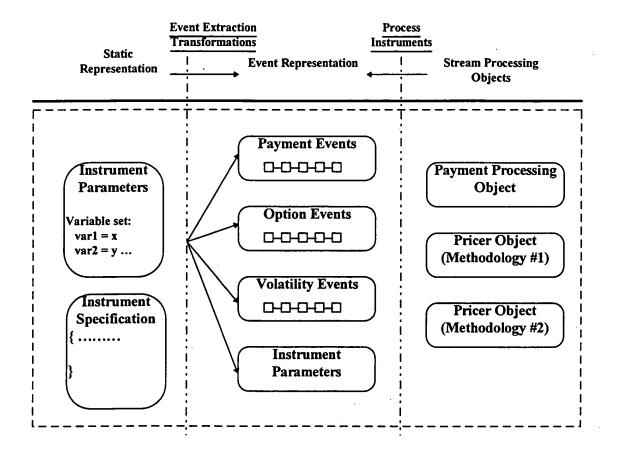


Figure 7 - Event Stream Representation

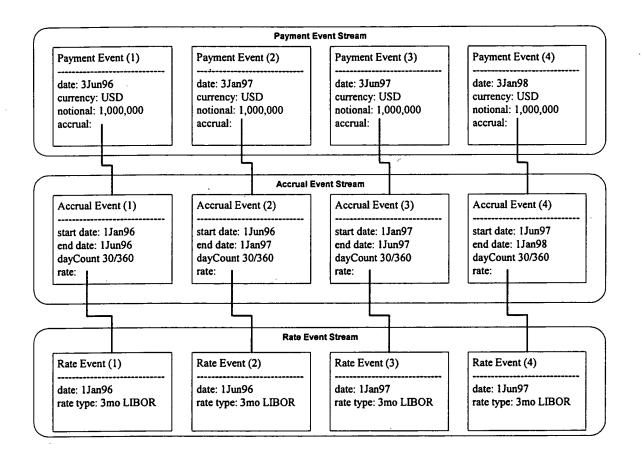


Figure 8 - Simple Swap Event Stream

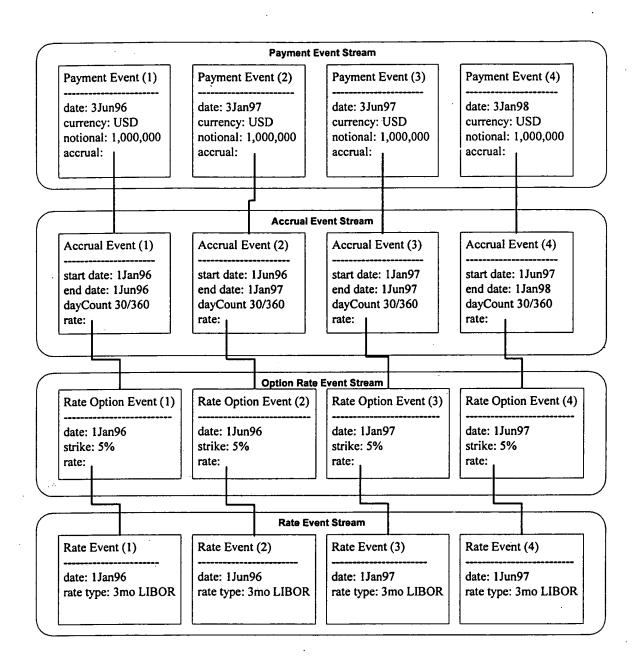


Figure 9 - Simple Option Event Stream Representation

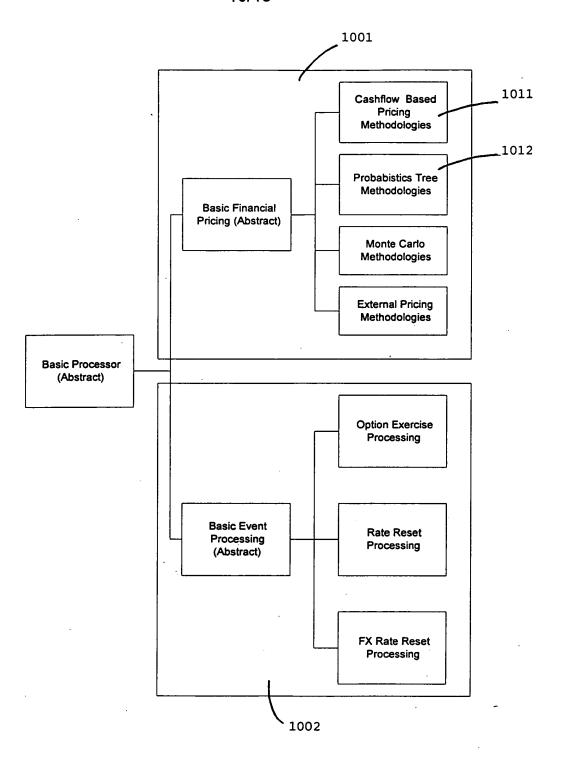


Figure 10 - Example Processing Class Hierarchy

Implemented Behaviour

InterestPaymentEvent Class Instance side:

valueEventInProcessor: aProcessor aProcessor valueInterestPaymentEvent: self

RateEvent Class Instance side:

valueEventInProcessor: aProcessor aProcessor valueRateEvent: self

Generic Processing Class Instance side:

valueInterestPaymentEvent: aPaymentEvent ...valuation action to go here...

valueRateEvent: aRateEventvaluation action to go here...

Figure 11 - Methods implemented on Event and Processing Classes

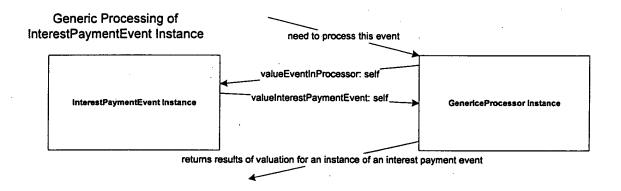


Figure 12 - Double Dispatch Processing for Interest Payment Event

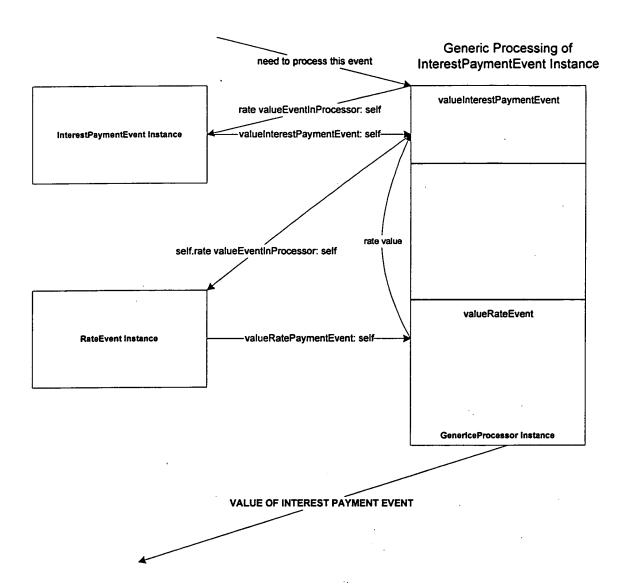


Figure 13 - Nested Double Dispatch

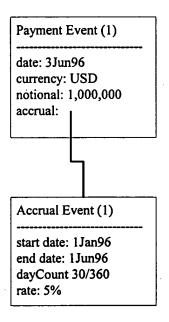


Figure 14 - Single Fixed Rate Payment Event